

WE CLAIM:

1. A system which transfers data across a network by means of a multi-window based GUI, comprising:

a remote server, having at least one window module, .NET application, .NET framework, .NET development tools, an HTML beginning tag, an HTML ending tag and at least one ASPX tag, wherein said remote server translates classes and objects into HTML/DHTML code by taking said ASPX tag and embedding HTML code to fit within said HTML beginning tag and said HTML ending tag and transferring said HTML/DHTML code across an electronic data network; and

at least one client system coupled to said remote server through said electronic data network, having a content retrieval module in communication with said remote server, wherein said at least one client system includes a windowed content manifestation environment; a web browser comprising at least one window module and at least one interactive menu module embedded in each said at least one window module, wherein each said at least one interactive menu module reacts to activation on client system and posts back to said remote server, notifying said .NET application through a raised event.

2. A system as in claim 1, further comprising a window icon docking system within said web browser within said client system, wherein said window icon docking system is in communication with each said at least one window module.

3. A system as in claim 2, further comprising a scrollable CSS div layer within said window icon docking system.

4. A system as in claim 1, further comprising at least one child component within said at least one window module.

5. A system as in claim 4, wherein said at least one child component is an interactive menu module.

6. A system as in claim 1, further comprising initialized menu items within said interactive menu module.

7. A system for transferring data across a network by means of a multi-window based GUI, further comprising:

5 a remote server, having at least one window module, .NET application, .NET framework, .NET development tools, an HTML beginning tag, an HTML ending tag and at least one ASPX tag, wherein said remote server translates classes and objects into HTML/DHTML code by taking said ASPX tag and embedding HTML code to fit within said HTML beginning tag and said HTML ending tag and transferring said HTML/DHTML code across an electronic data network; and

10 at least one client system coupled to said remote server through said electronic data network, having a content retrieval module in communication with said remote server, wherein said at least one client system includes a windowed content manifestation environment; a web browser comprising at least one window module, at least one interactive menu module embedded in each said at least one window module and a window icon docking system, wherein each said at least one  
15 interactive menu module reacts to activation on client system and posts back to said remote server, notifying said .NET application through a raised event, said

20 window icon docking system is present within said web browser within said client system, wherein said window icon docking system is in communication with each said at least one window module.

8. A system as in claim 7, further comprising a scrollable CSS div layer within said window icon docking system.

9. A system as in claim 7, wherein said interactive menu modules are embedded in each said at least one window module.

10. A system as in claim 7, further comprising a child component within said at least one window module.

11. A system as in claim 10, wherein said child component is a menu module.

12. A system as in claim 7, further comprising at least one initialized menu item within said at least one interactive menu module.

13. An object-oriented method of developing a software system, wherein at least one client system retrieves data from a remote server through a .NET environment using HTML/DHTML comprising the following steps:

- defining at least two object types;
- 5 creating at least one window object on said remote server to store programming code for generating dynamic HTML/DHTML context windows on said client system's said web browser;

creating at least one interactive menu object on said client system,  
wherein said at least one interactive menu object creates interactive menus from  
10 said programming code;

storing said programming code, from said client system, wherein said  
programming code provides for a set of steps that return selections from a user to  
applications on said remote server by means of a post back method;

acquiring said programming code from said remote server to said  
15 client system according to a set of steps using said at least two object types, said  
at least one window object and said at least one interactive menu object; and  
executing said programming code on said client system.

14. A method as defined in claim 13, further comprising the step of:  
creating at least one dock object on said client system, wherein said  
at least one dock object is a DHTML scrolling layer holding icons in communication  
with said dynamic HTML/DHTML context windows generated by said window  
5 object which have their state set to minimize.

15. A method as defined in claim 13, further comprising the step of  
including a scrollable CSS div layer within said at least one dock object.

16. A method as defined in claim 13, further comprising the step of  
creating a child component in said at least one window object.

17. A method as defined in claim 16, wherein said child component is an  
interactive menu object.

18. A method as defined in claim 13, further comprising the steps of:

modifying said at least one window object on said client system; and  
posting said modification back to said remote server at a  
predetermined time interval.

5

19. A method as defined in claim 18, wherein said step of modifying said  
at least one window object is chosen from the group of translation, resizing,  
collapsing, expansion, closing, minimizing, maximizing and restoring.

20. A method as defined in claim 13, further comprising the steps of:  
modifying said at least one window object on said remote server to  
create a modification; and  
posting said modification back to said remote server at a  
5 predetermined time interval.

21. A method as defined in claim 20, wherein said step of modifying said  
at least one window object is selected from the group of translation, resizing,  
collapsing, expansion, closing, minimizing, maximizing and restoring.

22. A method as defined in claim 13, further comprising the step of  
horizontally scrolling through content manifested within said at least one window  
object.

23. A method as defined in claim 13, further comprising the step of  
vertically scrolling through the content within said at least one window object by  
said user.

modifying said at least one window object on said client system; and  
posting said modification back to said remote server at a  
predetermined time interval.

5

19. A method as defined in claim 18, wherein said step of modifying said  
at least one window object is chosen from the group of translation, resizing,  
collapsing, expansion, closing, minimizing, maximizing and restoring.

20. A method as defined in claim 13, further comprising the steps of:  
modifying said at least one window object on said remote server to  
create a modification; and

posting said modification back to said remote server at a  
5 predetermined time interval.

21. A method as defined in claim 20, wherein said step of modifying said  
at least one window object is selected from the group of translation, resizing,  
collapsing, expansion, closing, minimizing, maximizing and restoring.

22. A method as defined in claim 13, further comprising the step of  
horizontally scrolling through content manifested within said at least one window  
object.

23. A method as defined in claim 13, further comprising the step of  
vertically scrolling through the content within said at least one window object by  
said user.

20 at least one window object, said at least one interactive menu object and said at least one dock object; and  
executing said programming code on said client system.

27. A method as defined in claim 26, further comprising the step of including a scrollable CSS div layer within said at least one dock object.

28. A method as defined in claim 26, further comprising the step of creating a child component in said at least one window object.

29. A method as defined in claim 28, wherein said child component is an interactive menu object.

30. A method as defined in claim 26, further comprising the steps of:  
modifying said at least one window object on said client system; and  
posting said modification back to said remote server at a predetermined time interval.

5

31. A method as defined in claim 30, wherein said step of modifying said at least one window object is chosen from the group of translation, resizing, collapsing, expansion, closing, minimizing, maximizing and restoring.

32. A method as defined in claim 26, further comprising the steps of:  
modifying said at least one window object on said remote server to create a modification; and  
posting said modification back to said remote server at a  
5 predetermined time interval.

33. A method as defined in claim 32, wherein said step of modifying said at least one window object is selected from the group of translation, resizing, collapsing, expansion, closing, minimizing, maximizing and restoring.

34. A method as defined in claim 26, further comprising the step of horizontally scrolling through content manifested within said at least one window object.

35. A method as defined in claim 26, further comprising the step of vertically scrolling through the content within said at least one window object by a user.

36. A method as defined in claim 26, further comprising the step of posting back to said remote server the resultant action of a raised event within said at least one menu object on said client system.

37. A method as defined in claim 26, further comprising the step of initializing items within said at least one menu object.